Amendments to the Claims:

Rewrite the claims as set forth below. This listing of claims replaces all prior versions and listings of claims in the application:

1. (currently amended) An apparatus for dual pass adaptive tessellation comprising:

a vertex grouper tessellator operably coupled to receive primitive information and an index list, wherein the index list is received from a memory device;

a shader processing unit coupled to the vertex grouper tessellator, wherein during a first pass, the shader processing unit receives primitive indices and an auto-index value for each of the ~~plurality of~~ primitive indices;

a plurality of vertex shader input staging registers operably coupled to the shader processing unit, each of the plurality of vertex shaders input staging registers coupled to one of a plurality of vertex shaders such that in response to a shader sequence output, the <u>plurality of</u> vertex shaders generate tessellation factors; and the tessellation factors are provided to the vertex grouper tessellator such that the vertex

grouper tessellator generates a per process vector output, a per primitive output and a per packet output.

2. (currently amended) The apparatus of claim 1 ~~further comprising:~~

~~the~~ <u>wherein the</u> tessellation factors <u>are</u> provided from the plurality of vertex shaders to a memory such that the memory ~~device~~ may provide the tessellation factors to the vertex grouper tessellator.

3. (currently amended) The apparatus of claim 1 ~~further comprising:~~

[[a]] wherein a plurality of coordinates are generated by the ~~tessellator in~~ vertex grouper tessellator in response to the plurality of tessellation ~~factor~~ factors.

4. (original) The apparatus of claim 3 wherein the plurality of coordinates are at least one of:

barycentric coordinates and tensor coordinates.

5. (currently amended) The apparatus of claim 3 ~~further comprising:~~

[[a]] wherein a plurality of tessellated vertices are generated by the plurality of vertex shaders in response to a plurality of control points and the plurality of coordinates.

6. (currently amended) The apparatus of claim 1 ~~further comprising~~ wherein a plurality of control points for higher order surfaces are generated by the plurality of vertex shaders based on a plurality of corner vertices of a plurality of primitives.

7. (currently amended) An apparatus for dual pass adaptive tessellation comprising:

a vertex ~~grouper-tessellator~~ grouper tessellator operably coupled to receive primitive information and a vertex index list from a memory device;

a shader processing unit coupled to the vertex grouper [[tesselator]] tessellator, wherein during a first pass, the shader processing unit receives primitive indices and an auto-index value for each of the ~~plurality of~~ primitive indices;

wherein the memory device is operably coupled to a plurality of vertex shaders such that tessellation factors are stored therein and wherein each of the plurality of vertex

shaders comprises a math processing unit coupled to the vertex ~~grouper-tessellator~~ grouper tessellator, the math processing unit including a plurality of input staging registers and a plurality of arithmetic logic units;

a control flow processor operatively coupled to the math processing unit wherein the control flow processor drives the math processing unit; ~~the memory device operably coupled to the plurality of vertex shaders such that the tessellation factors are stored therein;~~ and

wherein the vertex grouper tessellator retrieves the tessellation factors ~~in second~~ in a second pass such that the vertex grouper ~~tesselator~~ tessellator generates a per process vector output, a per primitive output and a per packet output.

8. (currently amended) The apparatus of claim 7 ~~further comprising:~~

[[a]] wherein a plurality of coordinates are generated by the vertex grouper ~~tesselator~~ tessellator in the second pass in response to the plurality of tessellation ~~factor~~ factors.

9. (original) The apparatus of claim 8 wherein the plurality of coordinates are at least one of:

barycentric coordinates and tensor coordinates.

10. (currently amended) The apparatus of claim 8 ~~further comprising:~~

[[a]] wherein a plurality of tessellated vertices are generated by the plurality of vertex shaders in response to a plurality of control points and the plurality of coordinates.

11. (currently amended) The apparatus of claim 7 ~~further comprising~~ wherein a plurality of control points for higher order surfaces are generated by the plurality

of vertex shaders based on a plurality of corner vertices of a plurality of primitives.

12. (currently amended) A method for dual pass adaptive tessellation comprising:

in a first pass:

receiving primitive information and an index list, wherein the index list is received from a memory device;

generating primitive indices from the primitive information and an auto-index value for each of the primitive indices;

generating a plurality of shader sequence outputs;

providing the shader sequence outputs to a plurality of vertex shader input staging [[registers:]] registers; and

generating a plurality of tessellation factors in response to the plurality of shader sequence outputs; and

in a second pass:

receiving the tessellation factors as a plurality of indices.

13. (currently amended) The method of claim 12 further comprising:

during the second pass:

generating an auto-index value for each of the plurality of indices;

generating a plurality of ~~bary-centric~~ coordinates based on the plurality of tessellation factors; and

computing a plurality of tessellated vertices by fetching a control point specified by the auto-index value for each of the plurality of indices.

14. (currently amended) The method of claim 13 wherein the coordinates are at least one of ~~bary-centric~~ <u>barycentric</u> coordinates and tensor coordinates.

15. (original) The method 12 further comprising:

providing the shader sequence outputs to a plurality of vertex shaders which are operably coupled to the vertex shader input staging registers.

16. (original) The method of claim 12 further comprising:

prior to the second pass, writing the plurality of tessellation factors to a memory device; and

during the second pass, receiving the tessellation factors from the memory device.

17. (original) The method of claim 16 wherein the tessellation factors are received from the memory device using a direct memory access.

18. (currently amended) A method for dual pass adaptive tessellation comprising:

in a first pass:

receiving vertex information and an index list, wherein the index list is received from a memory device;

generating primitive indices from the ~~primitive~~ <u>vertex</u> information and an auto-index value for each [[set]] of the primitive indices;

generating a plurality of shader sequence outputs;

providing the <u>plurality of</u> shader sequence outputs to a plurality of vertex shader input staging [[registers:]] <u>registers; and</u>

generating a plurality of tessellation factors in response to the <u>plurality of</u> shader sequence outputs; and

in a second pass:

receiving the plurality of tessellation factors as a plurality of indices;

generating an auto-index value for each of the plurality of indices;

generating a plurality of ~~bary-centric~~ barycentric coordinates based on the plurality of tessellation [[factors;; and]] factors; and

computing a plurality of tessellated vertices by fetching a control point specified by the auto-index value for each of the plurality of indices.

19. (currently amended) The method 18 further comprising:

providing the plurality of shader sequence outputs to a plurality of vertex shaders which are operably coupled to the vertex shader input staging registers.

20. (currently amended) The method of claim 18 further comprising:

prior to the second pass, writing the plurality of tessellation factors to [[a]] the memory device; and

during the second pass, receiving the tessellation factors from the memory device.

21. (currently amended) A method for dual pass adaptive tessellation comprising:

in a first pass:

receiving vertex information and an index list, wherein the index list is received from a memory device;

generating primitive indices from the ~~primitive~~ vertex information and an auto-index value for each [[set]] of the primitive indices;

generating a plurality of shader sequence outputs;

providing the plurality of shader sequence outputs to a plurality of vertex shader input staging [[registers:]] registers; and

generating a plurality of tessellation factors in response to the plurality of shader sequence outputs; and

in a second pass:

receiving the plurality of tessellation factors as a plurality of indices;

receiving a primitive type indicator; and

generating a set of coordinates based on the plurality of indices and the primitive type indicator.

22. (currently amended) The method of claim 21 further comprising:

during the second pass:

generating an auto-index value for each of the plurality of indices;

generating a plurality of ~~bary-centric~~ barycentic coordinates based on the tessellation ~~factors;; and~~ factors; and

computing a plurality of tessellated vertices by fetching a control point specified by the auto-index value for each of the plurality of indices.

23. (original) The method of claim 21 further comprising:

receiving a vertex reuse number; and

generating a plurality of vertices based on the vertex reuse number.

24. (currently amended) The method of claim 21 wherein the set of coordinates are parametric coordinates when the vertices define a plurality of tensor product

surfaces and the set of coordinates are ~~bary-centric~~ barycentric coordinates when the vertices define a plurality of triangular surfaces.

25. (original) The method 21 further comprising:

generating a plurality of sub-primitive vertices information.